## ****Weekly Report – Week 1: Go Language Fundamentals****

### ✅ ****Work Completed:****

1. **Go Setup & Tooling .**
   1. Installed Go, explored go mod, and used go fmt, go run, go build and go install.
   2. Familiarized myself with basic project structure.

Example:

*go mod init example.com/myapp*

*go run main.go*

1. **Variables, Types, Constants, Functions.**
2. Practiced declaration styles (var,:=,const)
3. Covered value vs reference types, default zero-values, type inference.
4. Created and used functions with multiple return values.

Example:

*func swap(a, b int) (int, int) {*

*return b, a*

*}*

1. **Pointers, Structs, Interfaces**
   1. Worked with pointers, memory references.
   2. Defined and used structs , embedded fields.
   3. Implemented basic interfaces and custom types.

Example:

*type Person struct {*

*Name string*

*Age int*

*}*

### ❓ ****Issues Faced:****

* Initially struggled with understanding interface implementations and pointer semantics, especially when passing to functions.
* Minor confusion with when to use value vs pointer receivers in methods.
* Learning how Go handles unused variables/errors in a strict way (e.g., \_ = err if not used).

**Tests Written:**

* Started writing unit tests using the built-in testing package.
* Used table-driven tests for better structure.
* Learned to run tests via go test ./… and check coverage.

**Sample Test Code:**

*func TestAdd(t \*testing.T) {*

*result := Add(2, 3)*

*if result != 5 {*

*t.Errorf("Expected 5, got %d", result)*

*}*

*}*

✅ Total tests written: 5  
✅ Table-driven test format used for max, min and pointer examples.

### ****Plan for Next Week:****

### Dive deeper into **Arrays, Slices, Maps, Control Flow**.

### Focus more on writing expressive **unit tests**.